class CWave

An Extension to the Microsoft Foundation Class Libraries (2.x)

Legal Terms

1.) You may copy and distribute the class Extensions, provided that all files remain intact and are distributed together.

2.) You may charge a fee for the physical act of transferring a copy and you may at your option offer a warranty protection in exchange for a fee.

3.) You may use and modify your copy or copies of the sources, thus forming a work based on them and copy and distribute such modifications or work under the terms of section 1, as long as you include notices stating that you changed the files and the date of any change.

NO WARRANTY

Because the files are licensed free of charge, there is no warranty for them, to the extent permitted by applicable law. The entire risk as to the quality and performance of the code is with you. Should the code prove defective, you assume the cost of all necessary servicing, repair or correction.
In no event unless required by applicable law or agreed to in writing will the copyright holder, or any other party who may modify and/or redistribute the code as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use the code (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a failure to operate with any other software), even if the copyright holder or other party has been advised of the possibility of such damages.

Description

The CWave class can be used to ease the handling of the Windows Multimedia Extensions when using the Microsoft Foundation Class Library in a custom application.
It does not require any 3rd party custom controls (.vbx).

Construction

CWave::CWave()
Creates an uninitialized (empty) wave object.

CWave::CWave(LPCSTR)
Creates a wave object that is initialized with a character string. The string may either contain the name of an external wave file or the name of an entry in the *sounds* section of the *win.ini* file.

CWave::CWave(UINT)
Creates a wave object that is initialized with the valued of a wave resource identifier.
When using this construction method, you must place the wave form binary in your application resource file.

CWave::CWave(CWave const&)
Creates (copies) a wave object from another wave object.

Destruction

CWave::~CWave()
Destroys a wave object, freeing all resources and unlocking all handles if necessary.

Loading Waves

BOOL CWave::LoadWave(LPCSTR)
Loads a wave form into a wave object.
The parameter may either be the name of an external wave file of the name of an entry in the *sounds* section of the *win.ini* file. The function will discard any wave form that had been attached to the wave object before.
The function returns *TRUE* if no error occurred initializing the wave form.

BOOL CWave::LoadWave(UINT)
Loads a wave form into a wave object.
The parameter specifies the identification number of a program resource wave form. The function will discard any wave form that had been attached to this object before.
The function returns *TRUE* if no error occurred initializing the wave form.

Assignment

void CWave::operator=(CWave const&)

Assigns a wave object from another one.

Output Operations

BOOL CWave::PlaySound(UINT = SND_SYNC)

Plays a wave attached to the object.
Returns *TRUE* if the output operation was successful.
The parameter for this function is similar to the *wFlags* parameter of the *::sndPlaySound(LPCSTR,WORD)* Windows function.
Its default value is *SND_SYNC* which causes the application to wait until all sound output is complete. There is no need to specify the *SND_MEMORY* flag for resident (resource) waves, as the CWave class takes care of this.
For other flag values refer to the *::sndPlaySound(LPCSTR,WORD)* function described in the *Microsoft Windows SDK Multimedia Programmer's Reference*.

void CWave::StopSound()

Stops any sound that is currently played.

Wave Status

BOOL CWave::IsLoaded() const

Returns *TRUE* if the object is initialized with a wave form.

BOOL CWave::IsResident() const

Returns *TRUE* if the object is initialized with a resident wave form (resource).

BOOL CWave::IsExternal() const

Returns *TRUE* if the object is initialized with an external wave form (*.wav file).

Debugging

virtual void AssertValid() const

In the debugging version of this class, this function overrides the function *CObject::AssertValid()*.
For more information about the *AssertValid()* member function please refer to the MFC documentation.

virtual void Dump(CDumpContext&) const

In the debugging version of this class, this function overrides the function *CObject::Dump(CDumpContect&)*.
For more information about the *Dump()* member function please refer to the MFC dosumentation.

Administration

If you find it difficult to use these sources, have bug reports or any suggestions for interesting extensions, please eMail me at

```
     Torsten_Neuer@sl.maus.de       (no mails >16K, please)
or
     75240.1522@compuserve.com
```

If possible, I try to help you with your problems or with any other topics related to these sources.
However, I am not promising you anything.